Alexnet模型由5个卷积层（其中一些后接3个池化Pooling 层 ），还有3个全连接层构成。AlexNet 跟 LeNet 结构类似，但使⽤了更多的卷积层和更⼤的参数空间来拟合⼤规模数据集 ImageNet。它是浅层神经⽹络和深度神经⽹络的分界线。

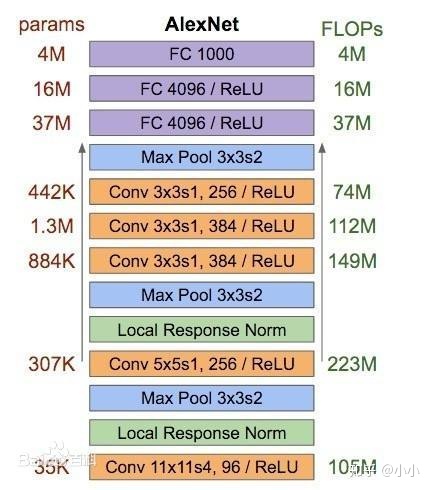
ReLU：增加非线性

LRN应用到深度神经网络中，这种横向抑制的目的是进行局部对比度增强，以便将局部特征在下一层得到表达。

卷积层负责边缘提取，也就是特征提取

池化层负责忽略细部特征，提取关键特征，也就是自适应滤波

全连接层负责做切割和判定，也就是识别



卷积核的个数=输出通道数

Prams:模型参数量。对于卷积层而言，公式为(k\_w\*k\_h\*c\_in)\*c\_out+c\_out

记卷积核的宽为k\_w，卷积核的高为k\_h，输入这一层的通道数是c\_in，这一层输出通道是c\_out

(k\_w\*k\_h\*c\_in)\*c\_out是权重的数量，最后加的c\_out是偏置的数量

对于全连接层而言，公式为(n\_in\*n\_out)+n\_out

假设这一层的输入拉伸成flat向量之后结点数为n\_in，这一层的输出结点数为n\_out

FLOPs:浮点运算次数，用来衡量模型的计算复杂度。计算FLOPs实际上是计算模型中乘法运算和加法运算的次数。

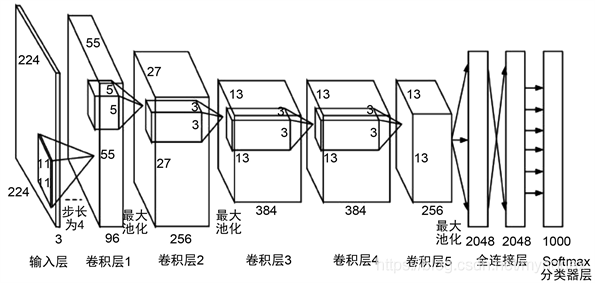
卷积层的FLOPs计算方法：卷积层所得到的每个feature map都是卷积核经过(k\_w\*k\_h\*c\_in)\*c\_out+c\_out次运算得到的，其中乘法运算次数为(k\_w\*k\_h\*c\_in)\*c\_out，加法运算次数为c\_out。假设这一层输出的feature map是 H \* W的，那这一层的浮点运算次数是[(k\_w\*k\_h\*c\_in)\*c\_out+c\_out]\*H\*W。可以看出来，浮点运算次数不仅取决于卷积核的大小和输入输出通道数，还取决于特征图的大小。

全连接层的FLOPs计算方法：全连接层的输出是经过(n\_in\*n\_out)+n\_out次计算得到的，所以全连接层的FLOPs和参数量是相同的。

另外：FLOPS(全部大写)：每秒浮点运算次数。用来衡量硬件的性能。

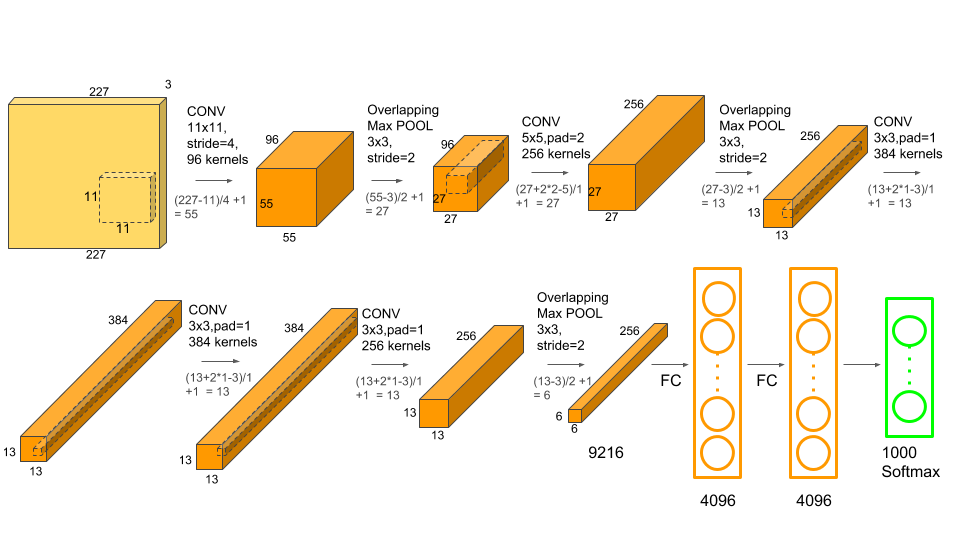
FLOPs：浮点运算次数，可以用来衡量算法/模型复杂度。

（有例子）



在AlexNet中，以第一个卷积层和第二个全连接层的计算为例。第一个卷积层卷积核是11\* 11 \* 3 \* 96的，如果在实现的时候加了偏置，那这一层的parameter为11\* 11 \* 3 \* 96 + 96 =34944，FLOPs为34944 \* 55 \* 55 =105705600。如果在实现的时候加了批量正则化而不使用偏置，则parameter为11\* 11 \* 3 \* 96=34848，FLOPs为34848 \* 55 \* 55=105415200。

第二个全连接层输入的结点数是2048，输出结点数也是2048，所以使用偏置时的parameter是2048\*2048+2048=4196352，FLOPs同为4196352，不使用偏置时的参数量时2048\*2048=4194304，FLOPs同为4194304



特点：

1、在每个卷积后面添加了ReLU激活函数，解决了Sigmoid的梯度消失问题，使收敛更快。

2、使用随机丢弃技术（dropout）选择性地忽略训练中的单个神经元，避免模型的过拟合（也使用数据增强防止过拟合）

3、添加了归一化LRN（Local Response Normalization，局部响应归一化）层，使准确率更高。

说到为什么要使用LRN就不得不提到神经生物学中的一个概念叫做 lateral inhibition（横向抑制），简单来讲就是兴奋的神经细胞抑制周围神经细胞的能力。应用到深度神经网络中，这种横向抑制的目的是进行局部对比度增强，以便将局部特征在下一层得到表达。

(LRN)的计算

LRN是通道层面，LRN是作用于卷积的一种优化，一般用于激活函数ReLU函数之后的一层。

4、重叠最大池化（overlapping max pooling），即池化范围 z 与步长 s 存在关系 z>s， 避免平均池化（average pooling）的平均效应。

AlexNet还应用了Overlapping（重叠池化），重叠池化就是池化操作在部分像素上有重合。池化核大小是n×n，步长是k，如果k=n，则是正常池化，如果 k<n, 则是重叠池化。官方文档中说明，重叠池化的运用减少了top-5和top-1错误率的0.4%和0.3%。重叠池化有避免过拟合的作用

AlexNet在fc6、fc7全连接层引入了drop out的功能。dropout是指在深度学习网络的训练过程中，对于神经网络单元，按照一定的概率（AlexNet是50%，这种情况下随机生成的网络结构最多）将其暂时从网络中丢弃（保留其权值），不再对前向和反向传输的数据响应。注意是暂时，对于随机梯度下降来说，由于是随机丢弃，故而相当于每一个mini-batch都在训练不同的网络，drop out可以有效防止模型过拟合，让网络泛化能力更强，同时由于减少了网络复杂度，加快了运算速度。还有一种观点认为drop out有效的原因是对样本增加来噪声，变相增加了训练样本

数据增强：在数据处理这部分作者提到过将每张图片处理为256××256的大小，但网络结构图中的输入却为224××224，这是因为作者在256××256大小的图片上使用了一个224××224的滑动窗口，将每个滑动窗口中的内容作为输入，这样就能将整个数据集扩大到原来的(256−224)×(256−224)=1024倍

conv1 阶段

输入数据：227×227×3

卷积核：11×11×3；步长：4；数量（也就是输出个数）：96

卷积后数据：55×55×96 （原图N×N，卷积核大小n×n，卷积步长大于1为k，输出维度是(N-n)/k+1）

relu1后的数据：55×55×96

Max pool1的核：3×3，步长：2

Max pool1后的数据：27×27×96

norm1：local\_size=5 （LRN(Local Response Normalization） 局部响应归一化）

最后的输出：27×27×96

conv2 阶段

输入数据：27×27×96

卷积核：5×5；步长：1；数量（也就是输出个数）：256

卷积后数据：27×27×256 （做了Same padding（相同补白），使得卷积后图像大小不变。）

relu2后的数据：27×27×256

Max pool2的核：3×3，步长：2

Max pool2后的数据：13×13×256 （（27-3）/2+1=13 ）

norm2：local\_size=5 （LRN(Local Response Normalization） 局部响应归一化）

最后的输出：13×13×256

conv2中使用了same padding，保持了卷积后图像的宽高不缩小。

conv3 阶段

输入数据：13×13×256

卷积核：3×3；步长：1；数量（也就是输出个数）：384

卷积后数据：13×13×384 （做了Same padding（相同补白），使得卷积后图像大小不变。）

relu3后的数据：13×13×384

最后的输出：13×13×384

conv3层没有Max pool层和norm层

conv4 阶段

输入数据：13×13×384

卷积核：3×3；步长：1；数量（也就是输出个数）：384

卷积后数据：13×13×384 （做了Same padding（相同补白），使得卷积后图像大小不变。）

relu4后的数据：13×13×384

最后的输出：13×13×38

conv4层也没有Max pool层和norm层

conv5 阶段

输入数据：13×13×384

卷积核：3×3；步长：1；数量（也就是输出个数）：256

卷积后数据：13×13×256 （做了Same padding（相同补白），使得卷积后图像大小不变。）

relu5后的数据：13×13×256

Max pool5的核：3×3，步长：2

Max pool2后的数据：6×6×256 （（13-3）/2+1=6 ）

最后的输出：6×6×256

conv5层有Max pool，没有norm层

fc6 阶段

输入数据：6×6×256

全连接输出：4096×1

relu6后的数据：4096×1

drop out6后数据：4096×1

最后的输出：4096×1

fc7 阶段

输入数据：4096×1

全连接输出：4096×1

relu7后的数据：4096×1

drop out7后数据：4096×1

最后的输出：4096×1

fc8阶段

输入数据：4096×1

全连接输出：1000

fc8输出一千种分类的概率

整体来看，AlexNet的卷积核从11到5再到3不断变小，而feature map也通过重叠式max pool在第1、2、5层折半式缩小，到第5个卷积层后，图像特征已经提炼得足够充分，便用两个全连接层和一个softmax层组合得出最终的分类结果。